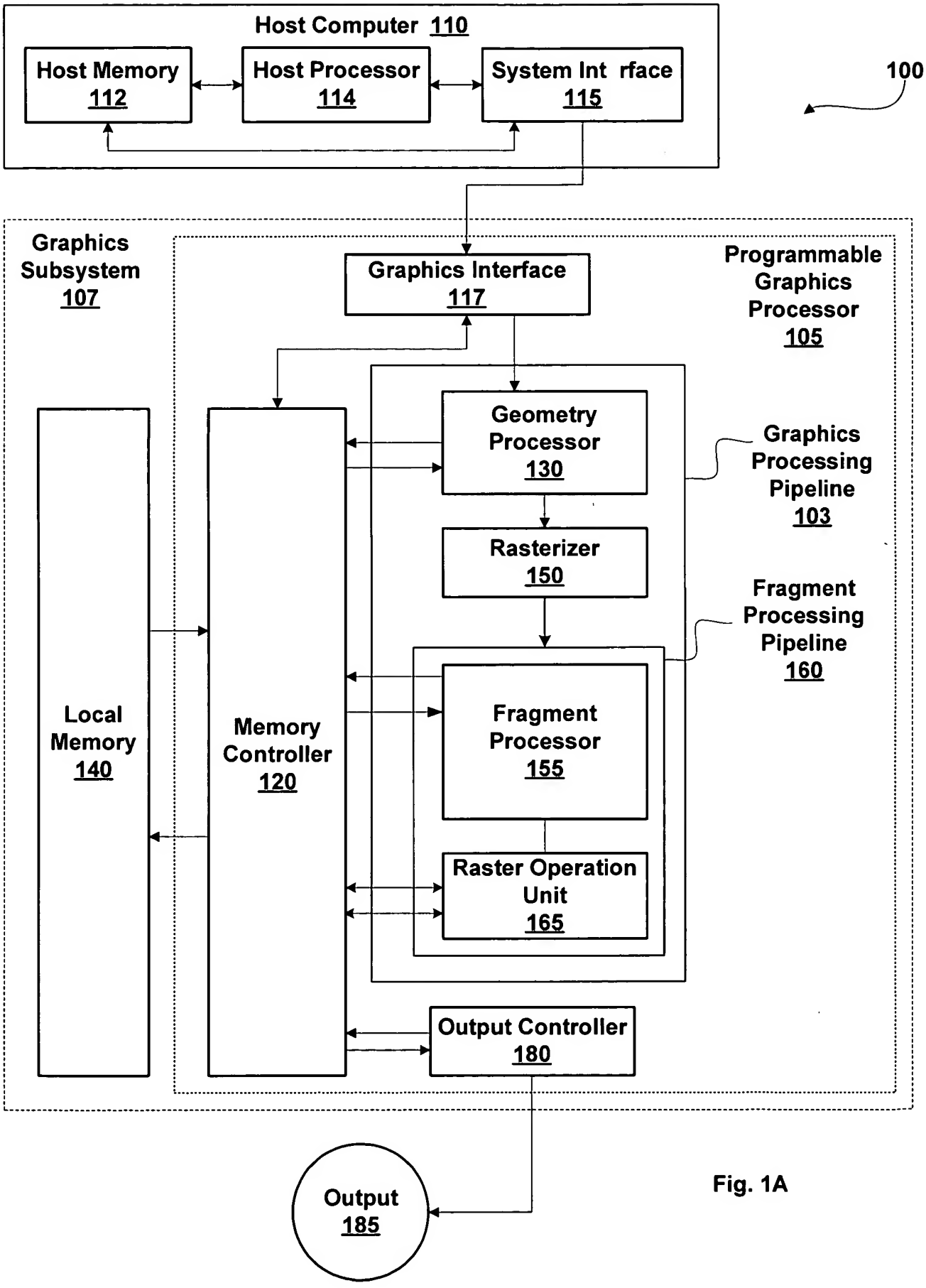
Host Computer 110
Host Memory 112
Host Processor 114
System Int rface 115
100
Graphics Subsystem 107
Graphics Interface 117
Programmable Graphics Processor 105
Geometry Processor 130
Graphics Processing Pipeline 103
Rasterizer 150
Fragment Processing Pipeline 160
Local Memory 140
Memory Controller 120
Fragment Processor 155
Raster Operation Unit 165
Output Controller 180
Output 185

Fig. 1A

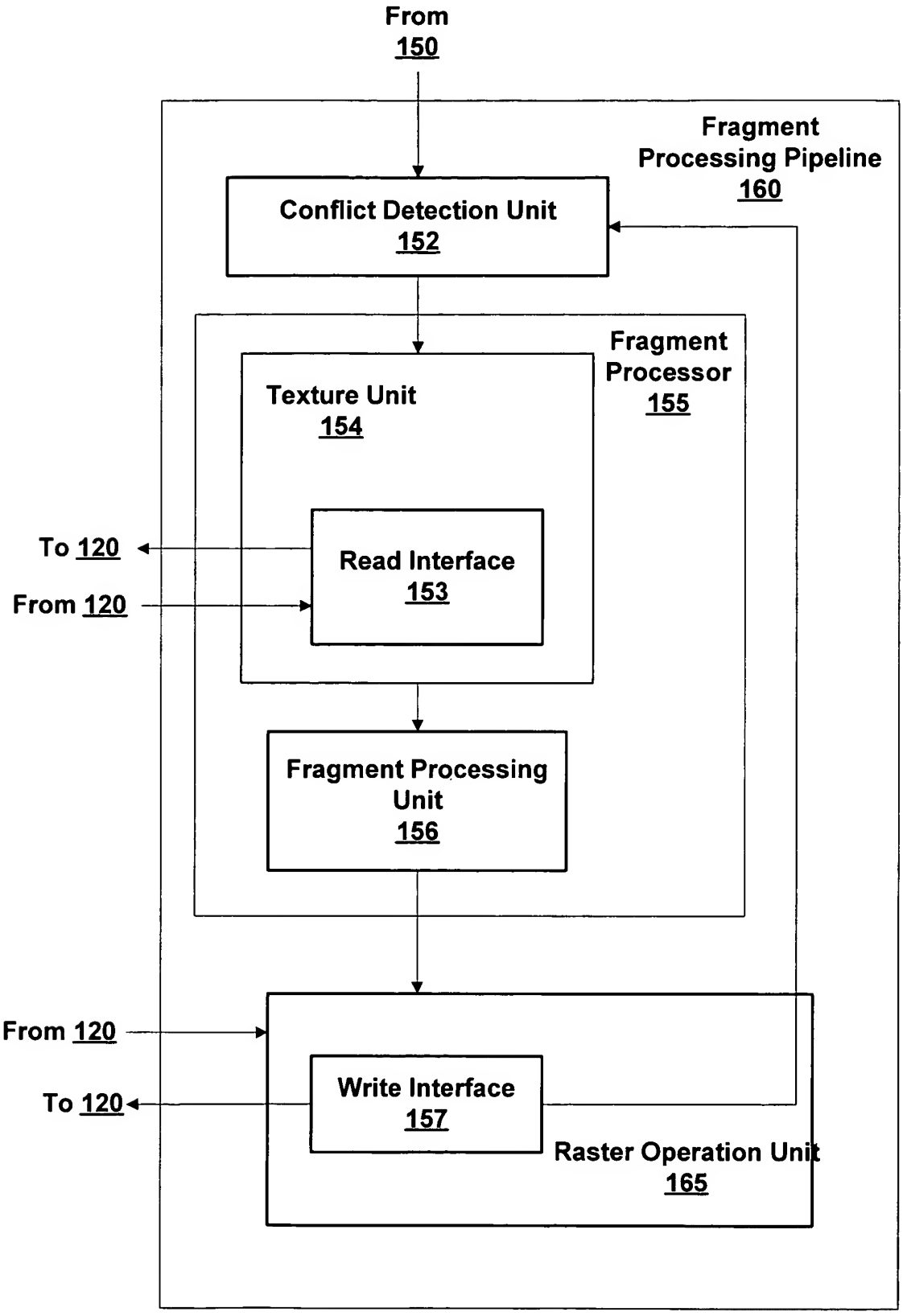
From
150
Fragment
Processing Pipeline
160
Conflict Detection Unit
152
Fragment
Processor
155
Texture Unit
154
To 120
From 120
Read Interface
153
Fragment Processing
Unit
156
From 120
To 120
Write Interface
157
Raster Operation Unit
165

Fig. 1B

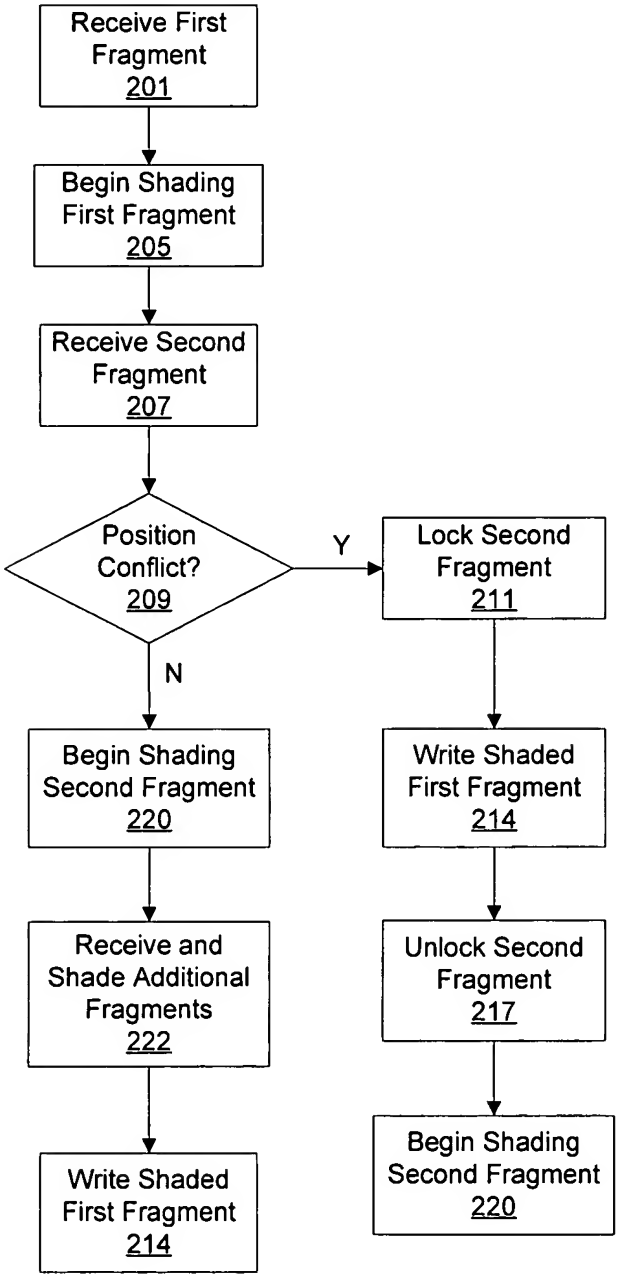
Receive First Fragment 201
Begin Shading First Fragment 205
Receive Second Fragment 207
Position Conflict? 209
Y
N
Lock Second Fragment 211
Write Shaded First Fragment 214
Unlock Second Fragment 217
Begin Shading Second Fragment 220
Begin Shading Second Fragment 220
Receive and Shade Additional Fragments 222
Write Shaded First Fragment 214

Fig. 2A

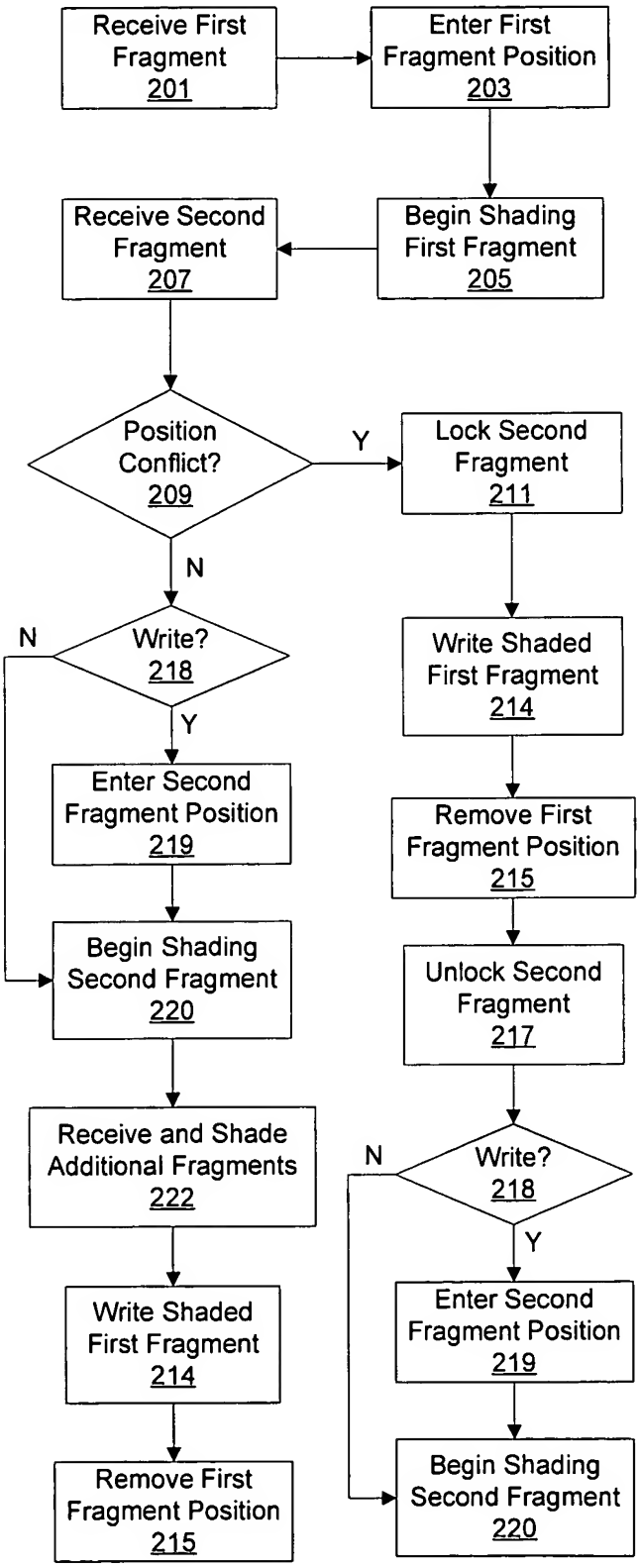
Receive First Fragment 201
Enter First Fragment Position 203
Begin Shading First Fragment 205
Receive Second Fragment 207
Position Conflict? 209
Y
Lock Second Fragment 211
N
Write? 218
N
Y
Enter Second Fragment Position 219
Begin Shading Second Fragment 220
Receive and Shade Additional Fragments 222
Write Shaded First Fragment 214
Remove First Fragment Position 215
Write Shaded First Fragment 214
Remove First Fragment Position 215
Unlock Second Fragment 217
Write? 218
N
Y
Enter Second Fragment Position 219
Begin Shading Second Fragment 220

Fig. 2B

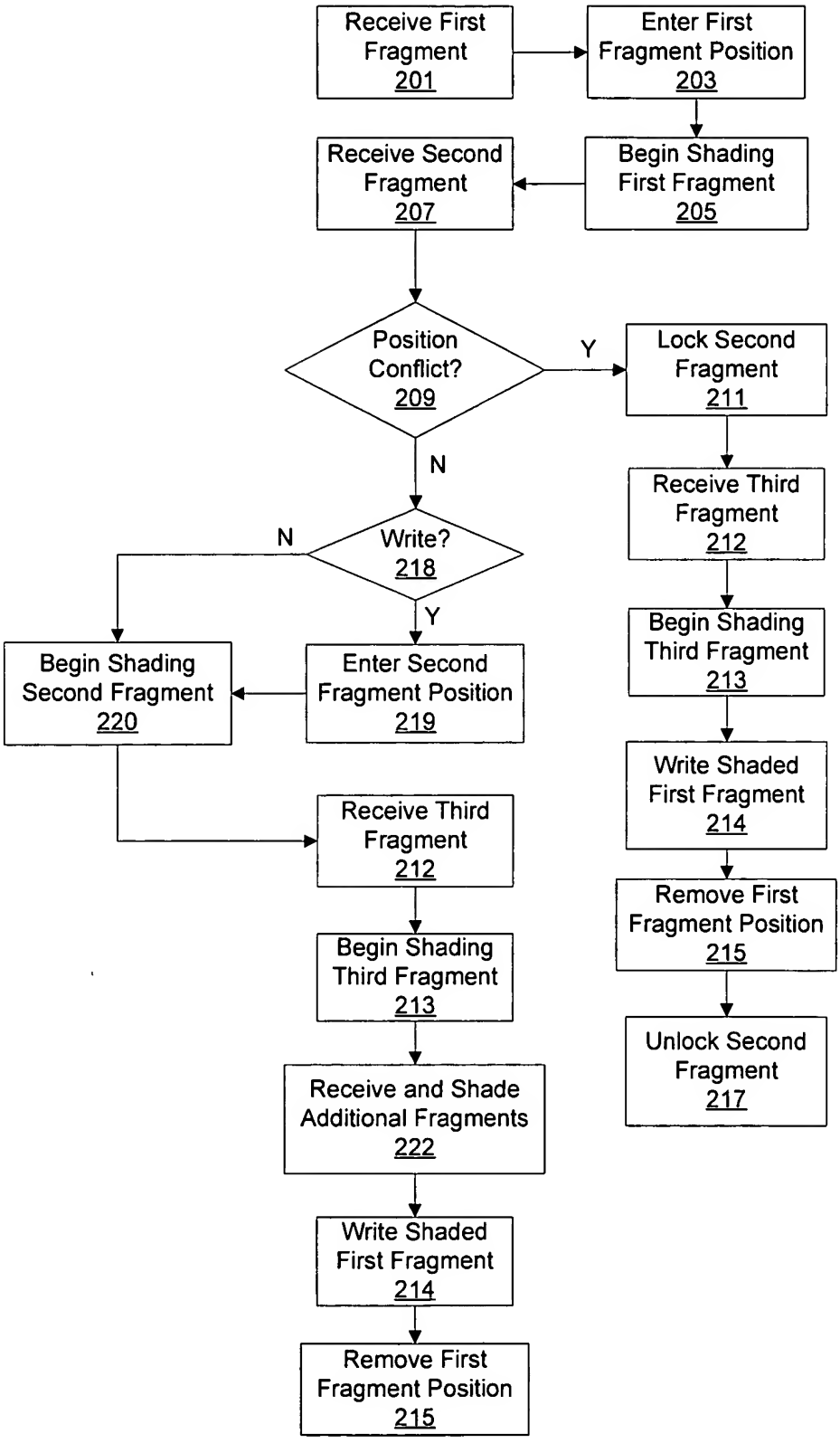
Receive First Fragment 201
Enter First Fragment Position 203
Begin Shading First Fragment 205
Receive Second Fragment 207
Position Conflict? 209
Y
Lock Second Fragment 211
N
Receive Third Fragment 212
Write? 218
N
Y
Begin Shading Third Fragment 213
Begin Shading Second Fragment 220
Enter Second Fragment Position 219
Write Shaded First Fragment 214
Receive Third Fragment 212
Remove First Fragment Position 215
Begin Shading Third Fragment 213
Unlock Second Fragment 217
Receive and Shade Additional Fragments 222
Write Shaded First Fragment 214
Remove First Fragment Position 215

Fig. 2C

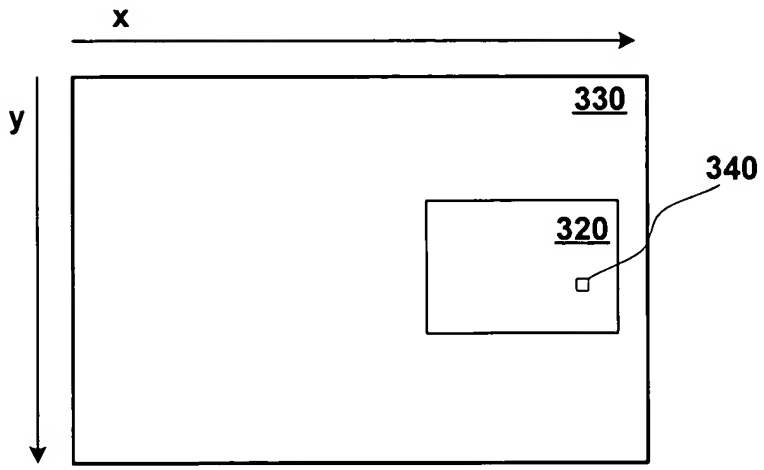

Fig. 3A

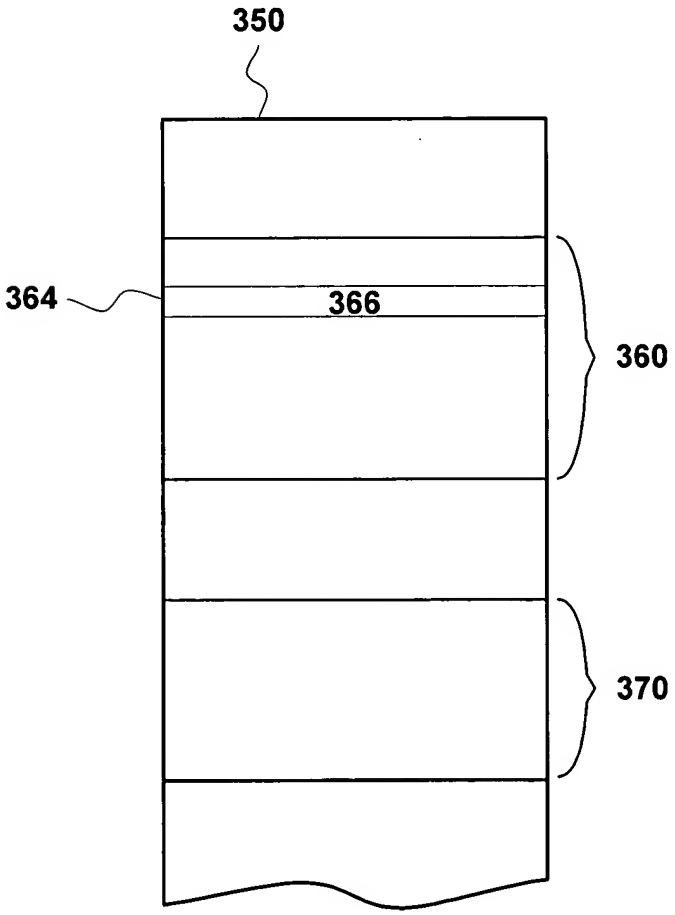

Fig. 3B

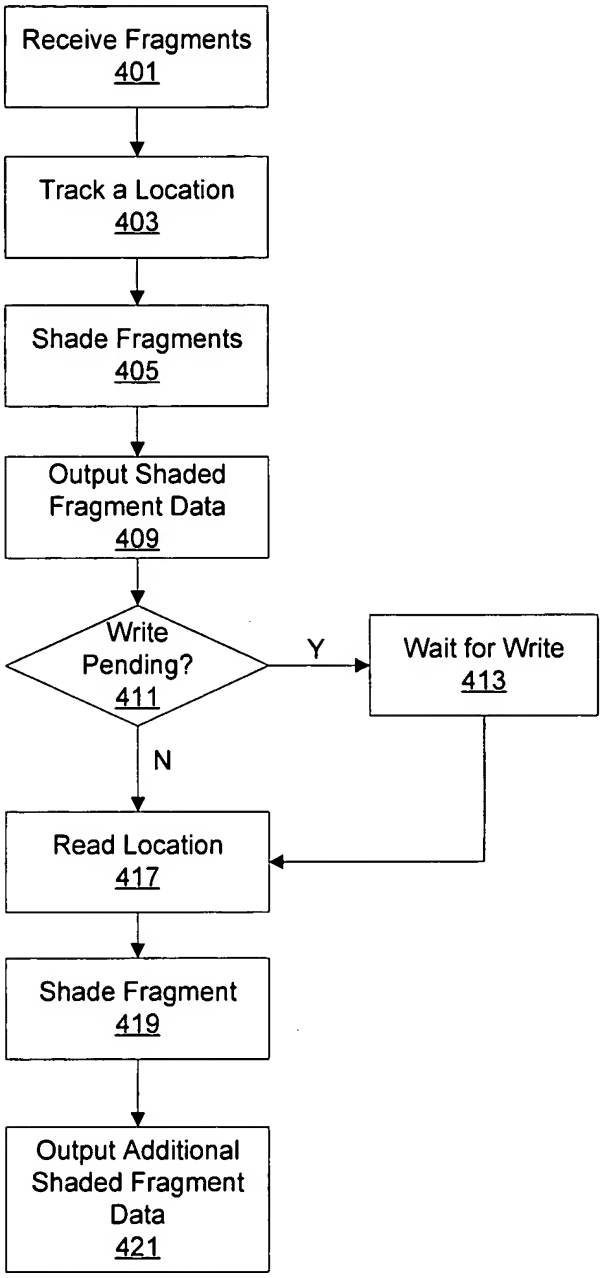
Receive Fragments
401
Track a Location
403
Shade Fragments
405
Output Shaded Fragment Data
409
Write Pending?
411
Y
Wait for Write
413
N
Read Location
417
Shade Fragment
419
Output Additional Shaded Fragment Data
421

Fig. 4A

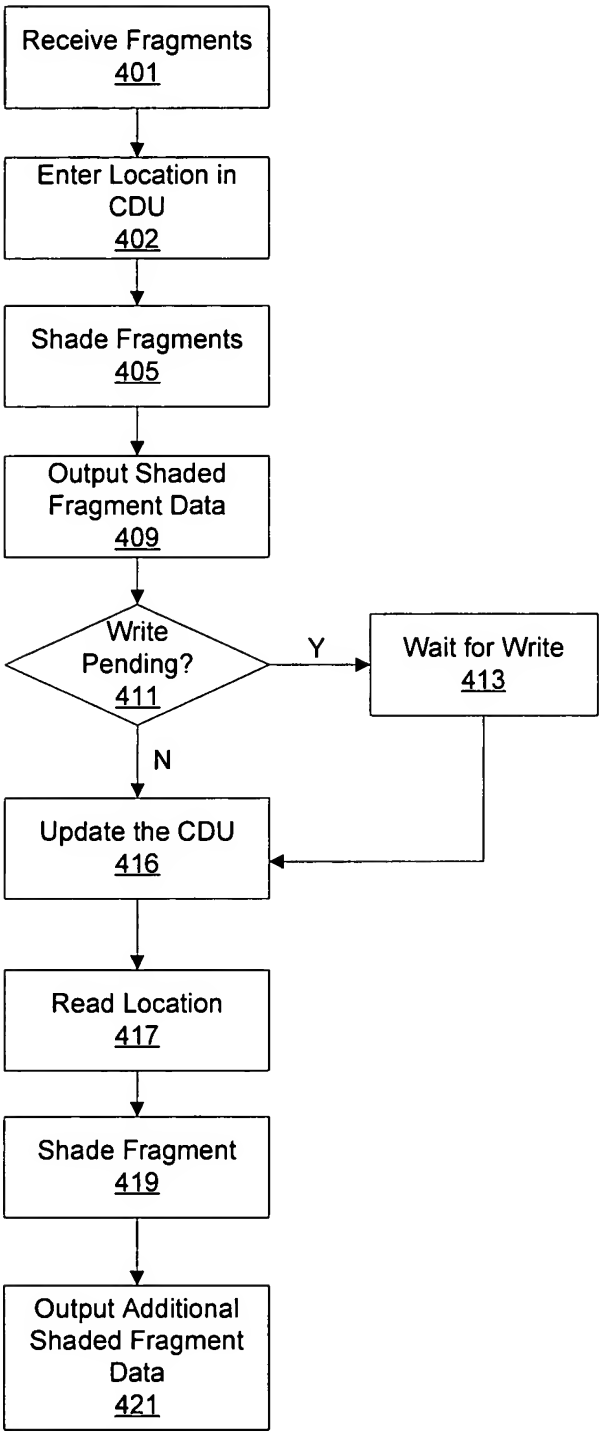
Receive Fragments
401
Enter Location in CDU
402
Shade Fragments
405
Output Shaded Fragment Data
409
Write Pending?
411
Y
Wait for Write
413
N
Update the CDU
416
Read Location
417
Shade Fragment
419
Output Additional Shaded Fragment Data
421

Fig. 4B

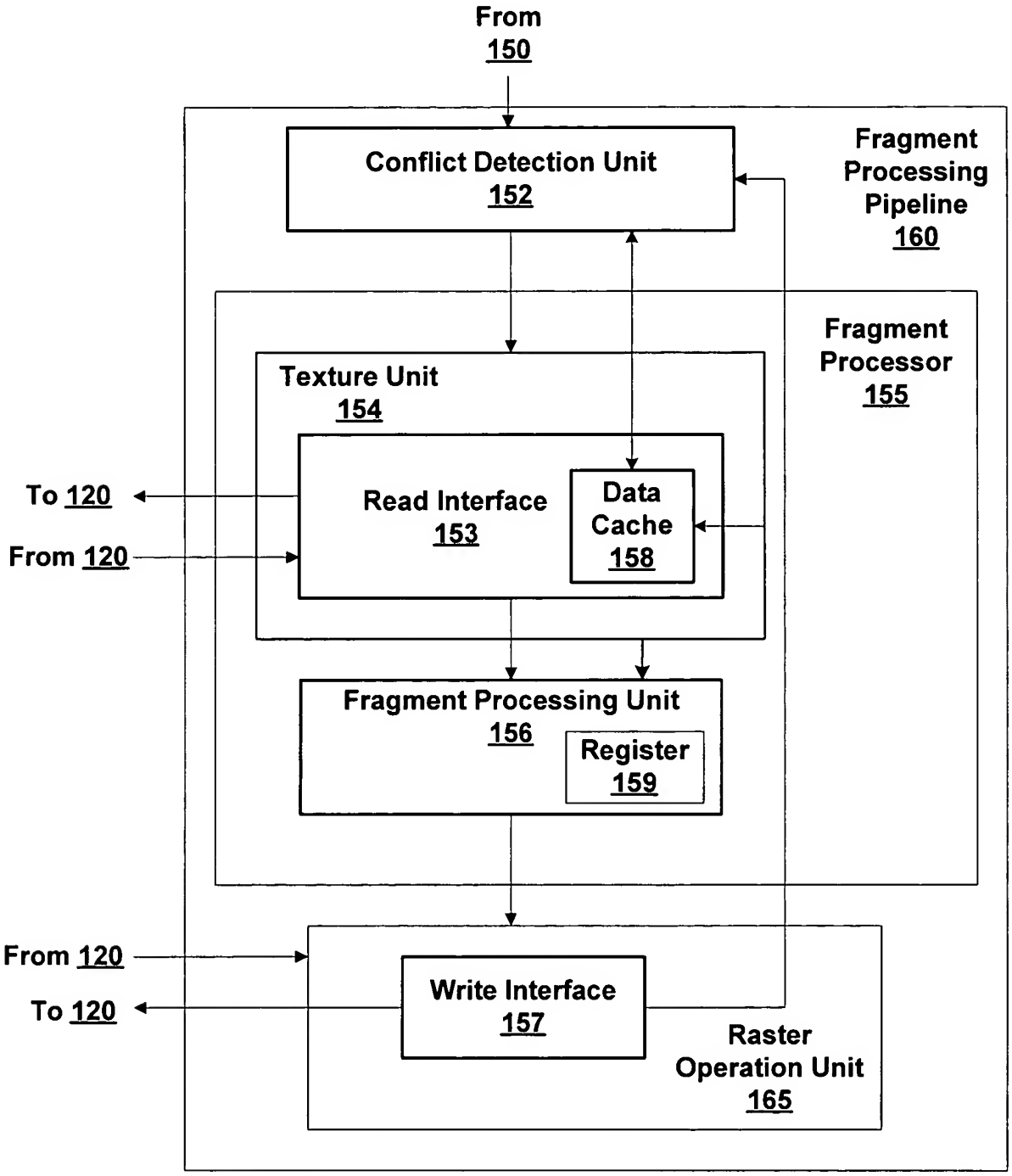
From
150
Conflict Detection Unit
152
Fragment
Processing
Pipeline
160
Fragment
Processor
155
Texture Unit
154
Read Interface
153
Data
Cache
158
To 120
From 120
Fragment Processing Unit
156
Register
159
From 120
To 120
Write Interface
157
Raster
Operation Unit
165

Fig. 5

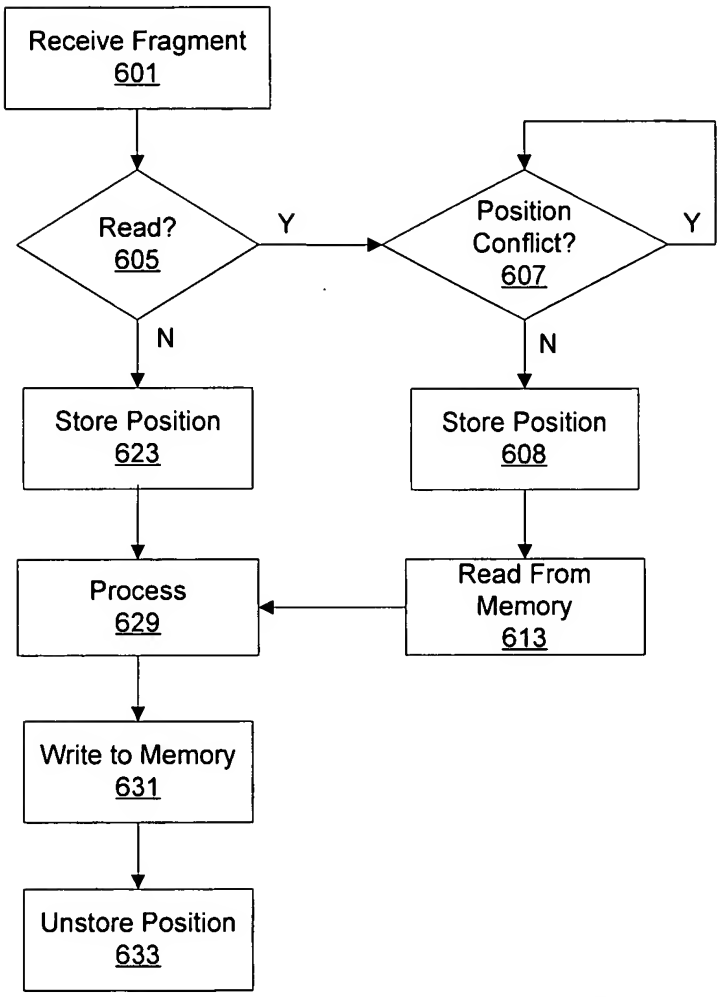
Receive Fragment
601
Read?
605
Y
Position
Conflict?
607
Y
N
N
Store Position
623
Store Position
608
Process
629
Read From
Memory
613
Write to Memory
631
Unstore Position
633

Fig. 6A

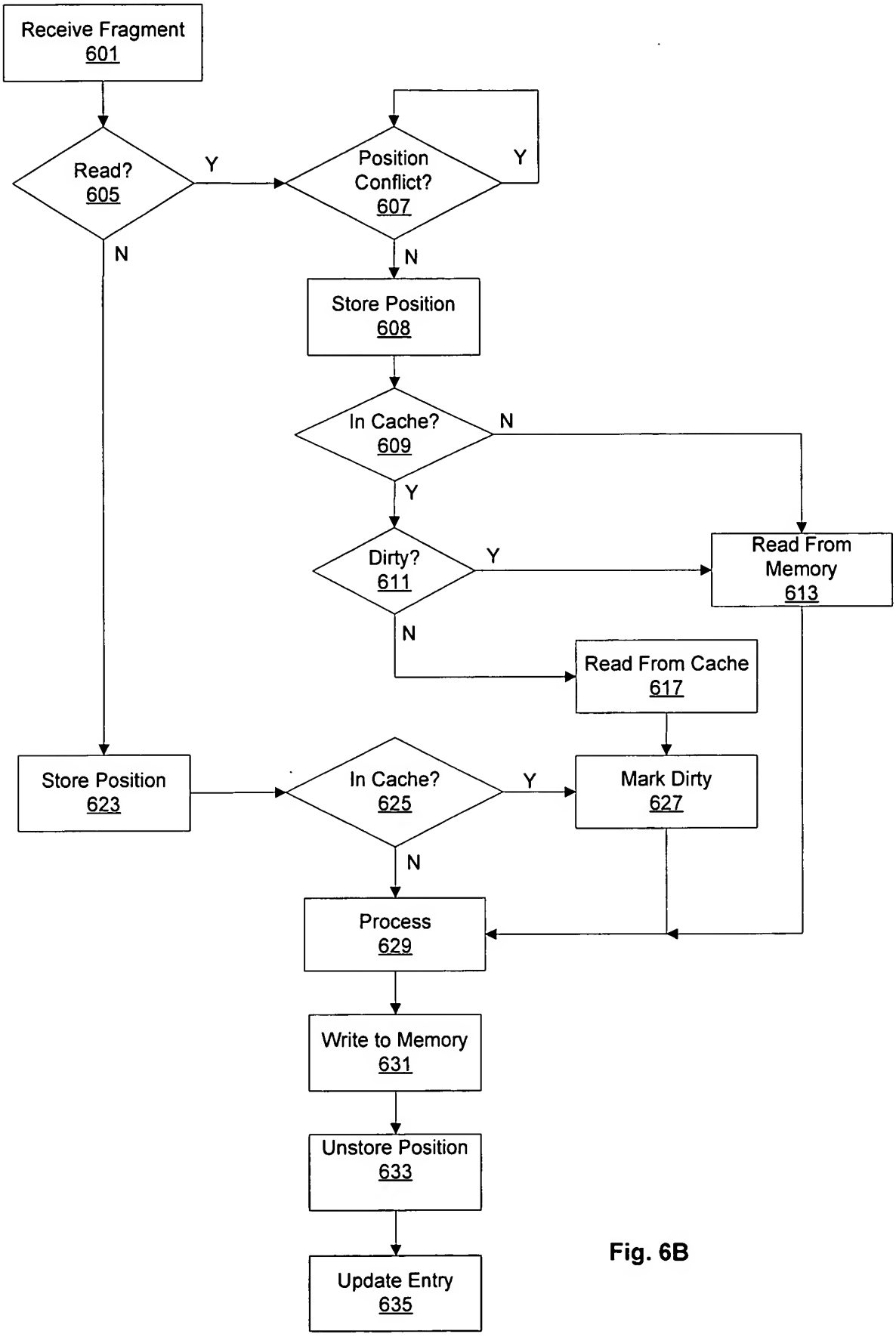
Receive Fragment
601
Read?
605
Y
N
Position
Conflict?
607
Y
N
Store Position
608
In Cache?
609
N
Y
Dirty?
611
Y
N
Read From
Memory
613
Read From Cache
617
Store Position
623
In Cache?
625
Y
N
Mark Dirty
627
Process
629
Write to Memory
631
Unstore Position
633
Update Entry
635

Fig. 6B

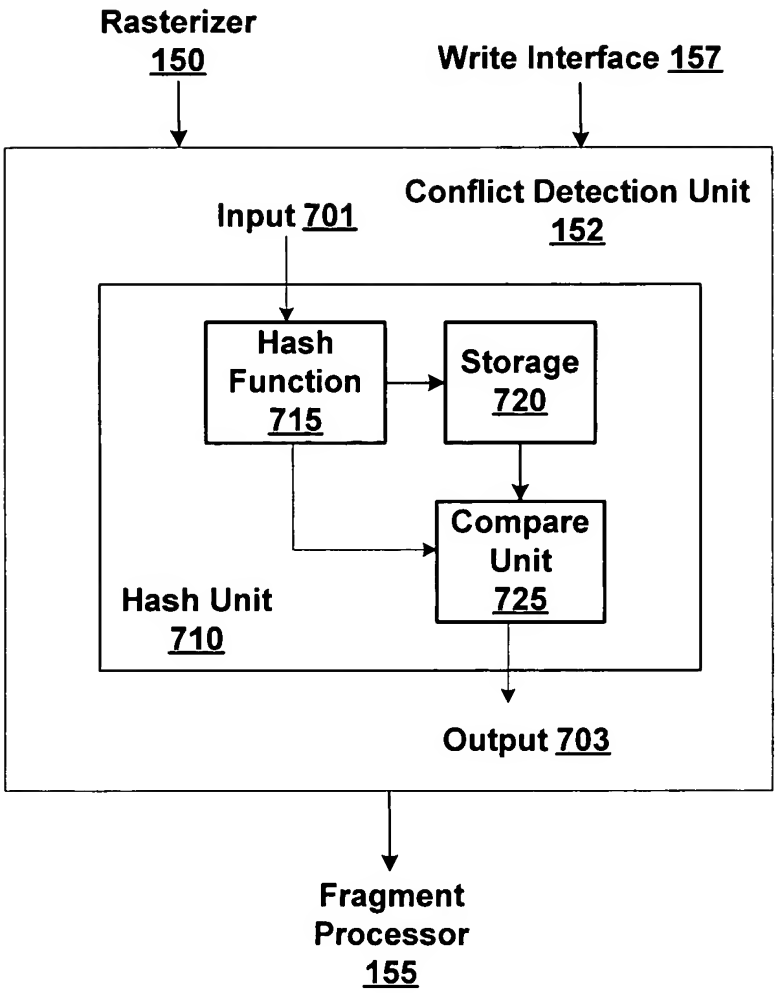
Rasterizer
150
Write Interface 157
Conflict Detection Unit
152
Input 701
Hash
Function
715
Storage
720
Compare
Unit
725
Hash Unit
710
Output 703
Fragment
Processor
155

Fig. 7

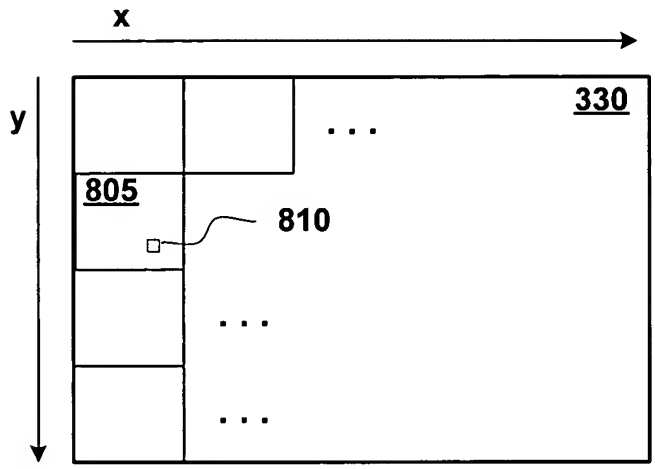

Fig. 8A

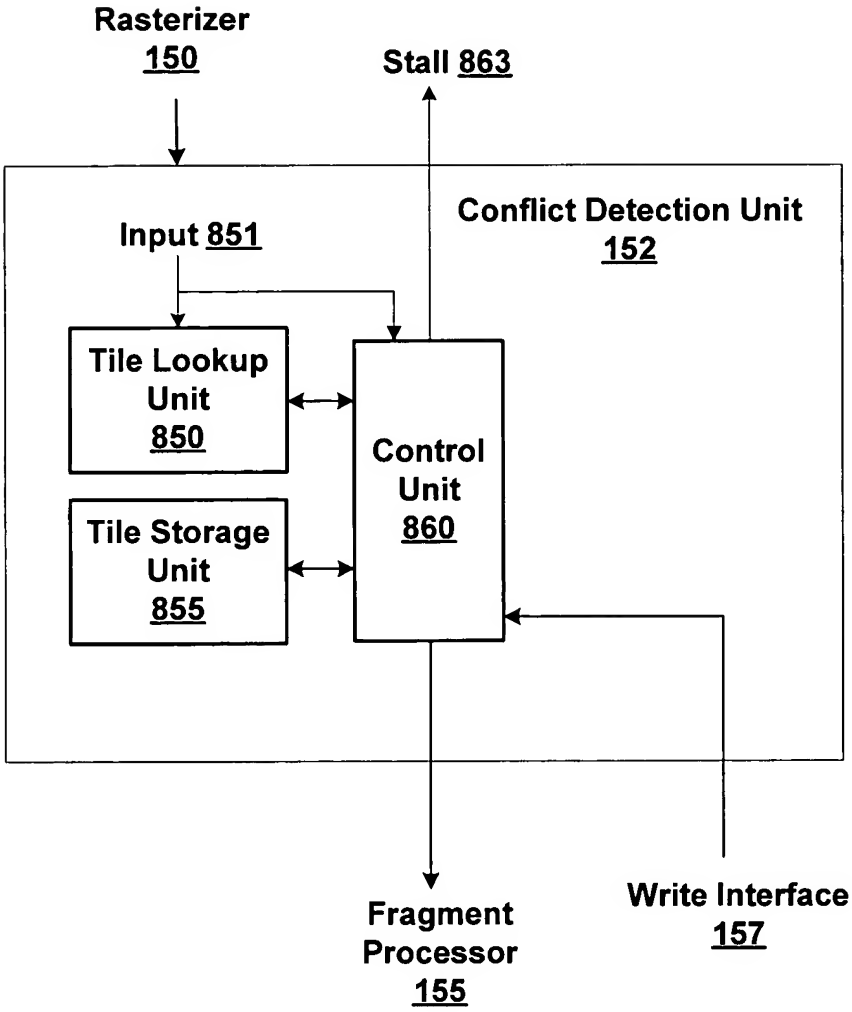
Rasterizer
150
Stall 863
Conflict Detection Unit
152
Input 851
Tile Lookup
Unit
850
Tile Storage
Unit
855
Control
Unit
860
Fragment
Processor
155
Write Interface
157

Fig. 8B

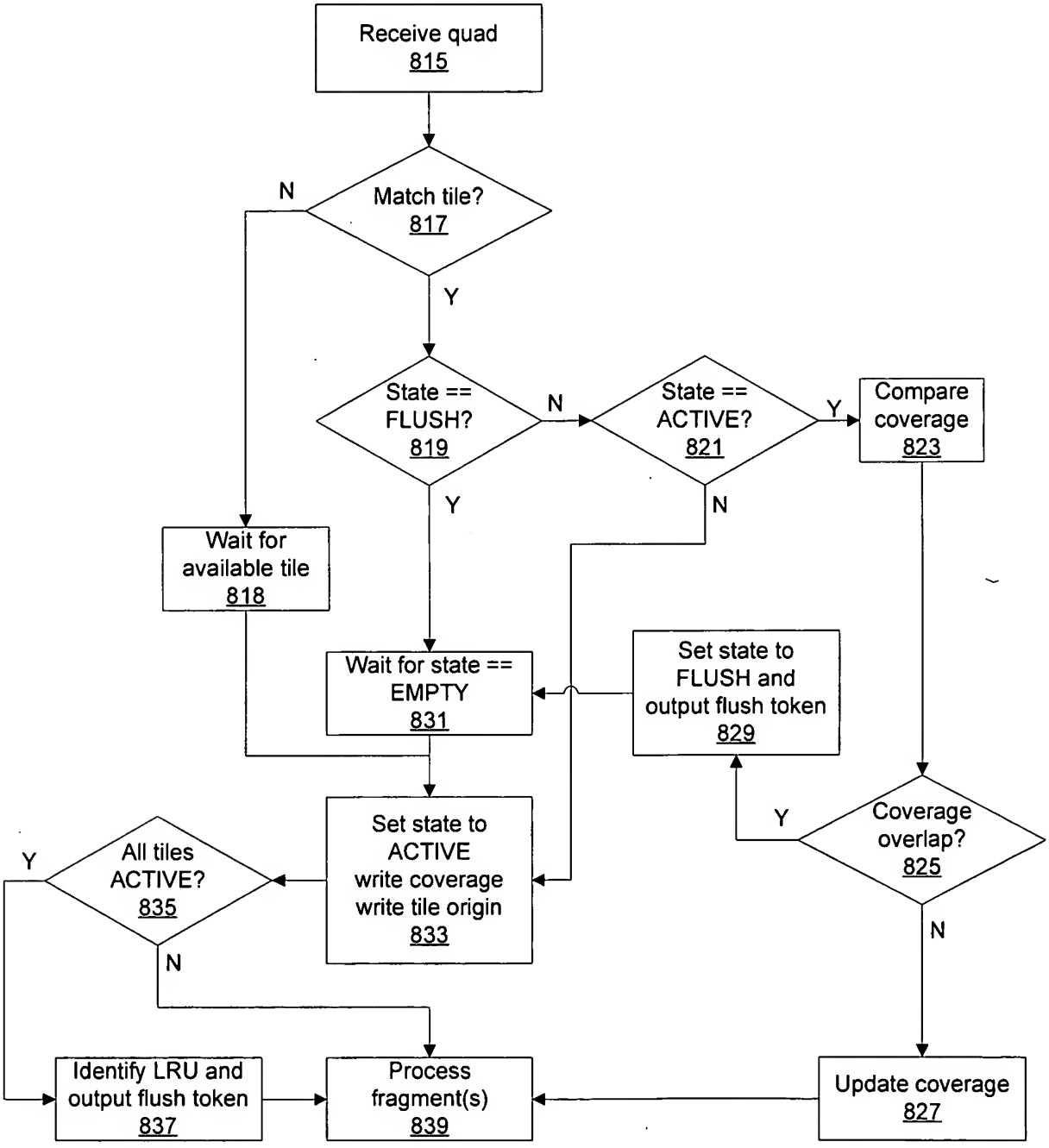
Receive quad
815
Match tile?
817
N
Y
State == FLUSH?
819
N
State == ACTIVE?
821
Y
Compare coverage
823
Y
N
Wait for available tile
818
Wait for state == EMPTY
831
Set state to FLUSH and output flush token
829
Set state to ACTIVE write coverage write tile origin
833
All tiles ACTIVE?
835
Y
N
Coverage overlap?
825
Y
N
Identify LRU and output flush token
837
Process fragment(s)
839
Update coverage
827

Fig. 8C

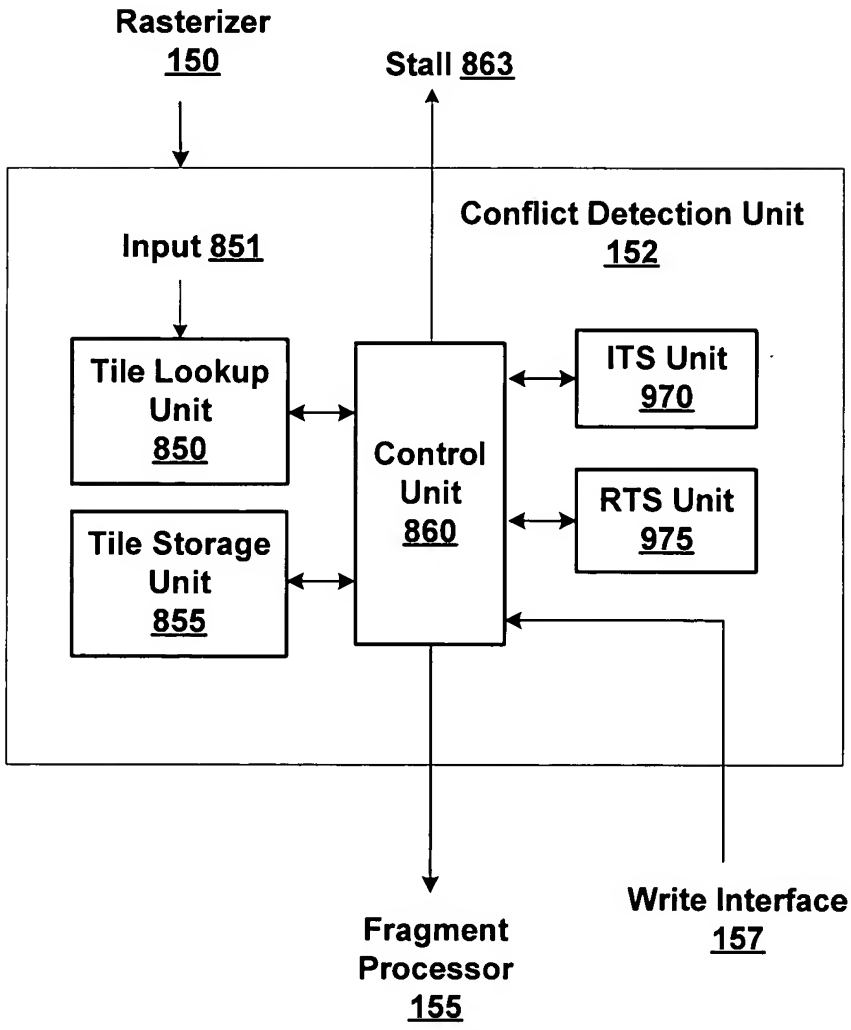
Rasterizer
150
Stall 863
Conflict Detection Unit
152
Input 851
Tile Lookup
Unit
850
Tile Storage
Unit
855
Control
Unit
860
ITS Unit
970
RTS Unit
975
Fragment
Processor
155
Write Interface
157

Fig. 9A

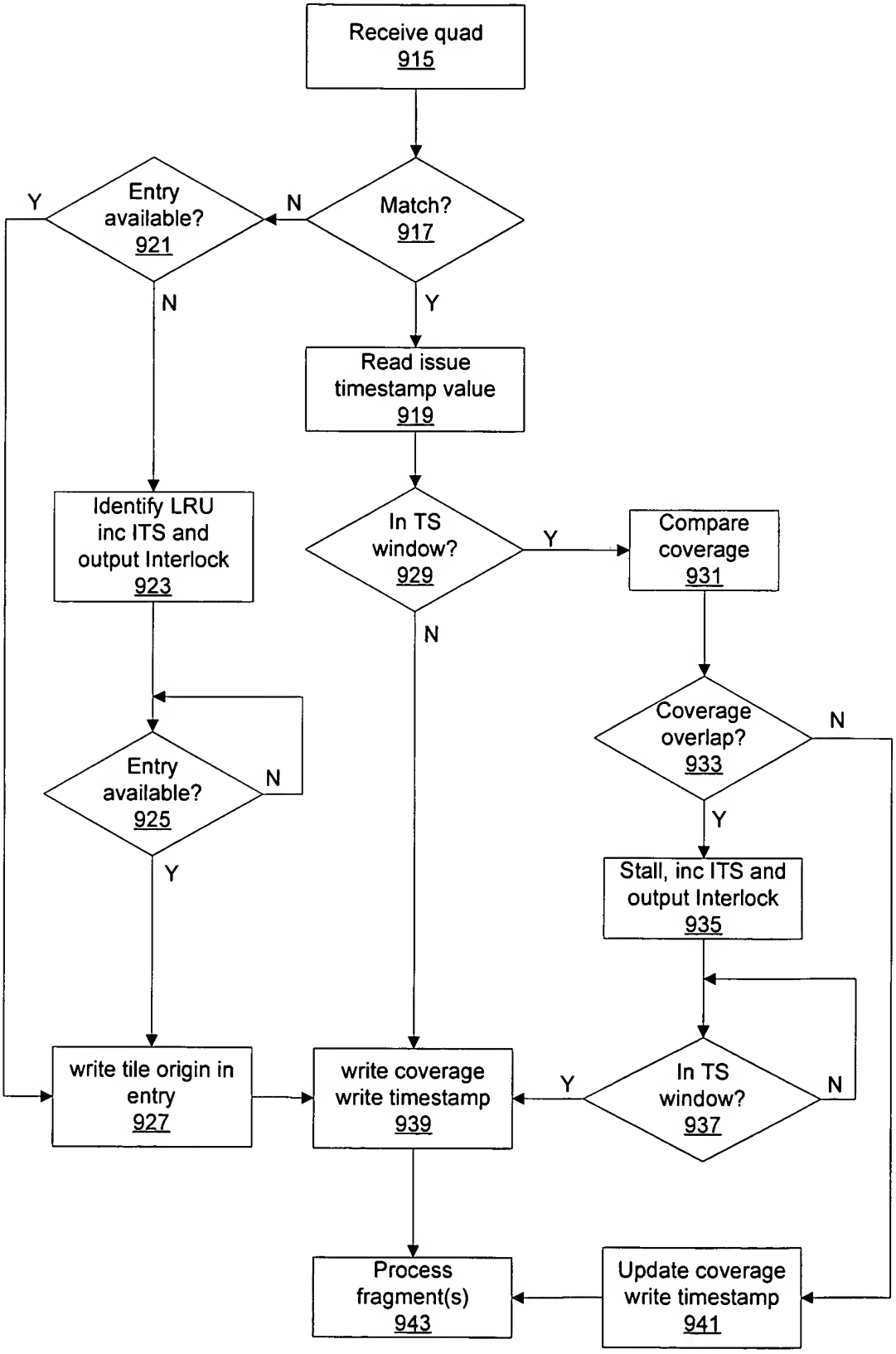
Receive quad
915
Match?
917
N
Entry available?
921
Y
N
Y
Read issue timestamp value
919
Identify LRU inc ITS and output Interlock
923
In TS window?
929
Y
Compare coverage
931
N
Coverage overlap?
933
N
Y
Entry available?
925
N
Y
Stall, inc ITS and output Interlock
935
write tile origin in entry
927
write coverage write timestamp
939
Y
In TS window?
937
N
Process fragment(s)
943
Update coverage write timestamp
941

Fig. 9B

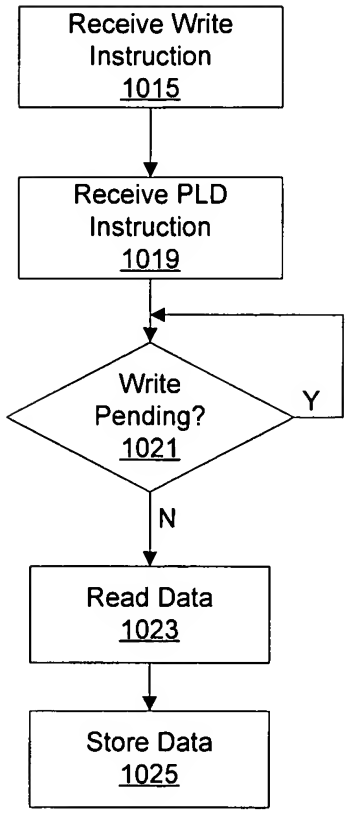

Fig. 10A

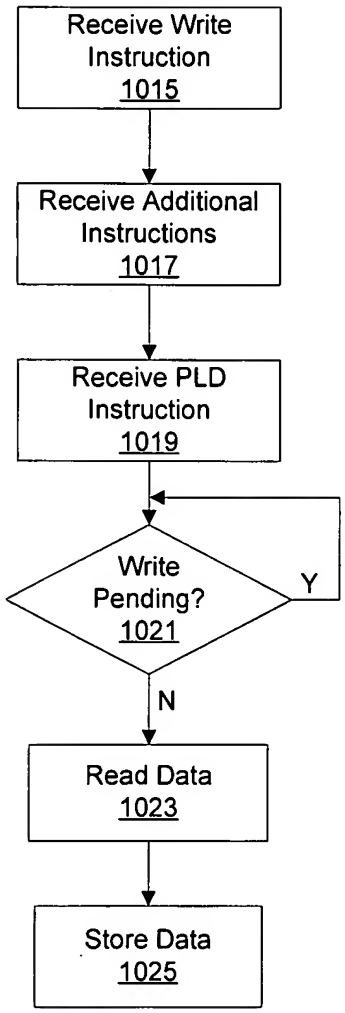

Fig. 10B

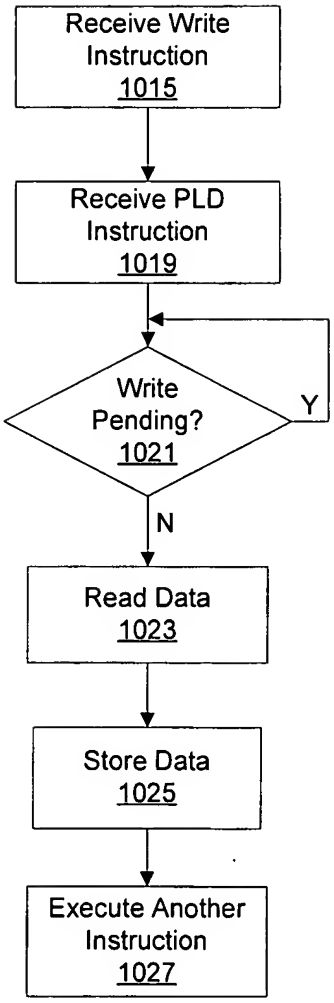

Fig. 10C

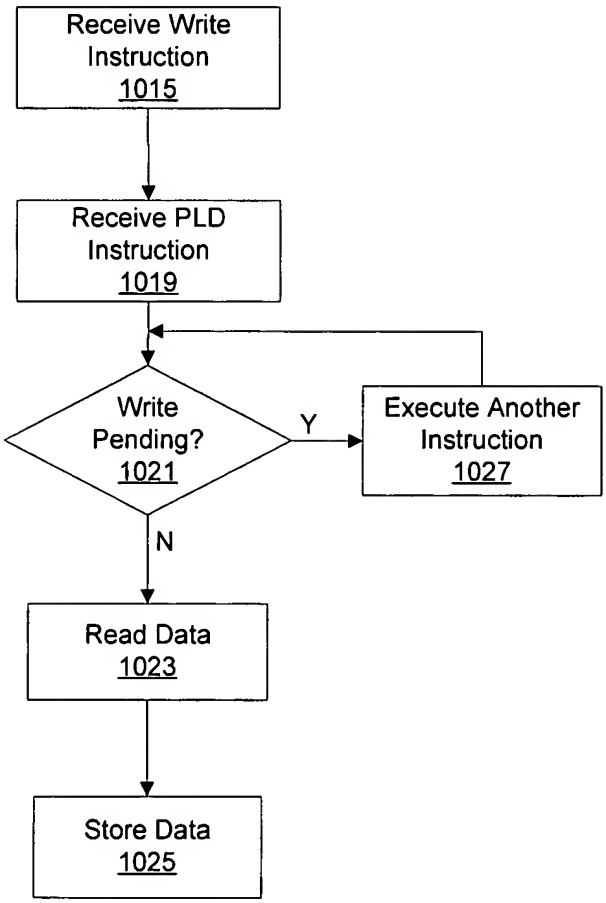

Fig. 10D

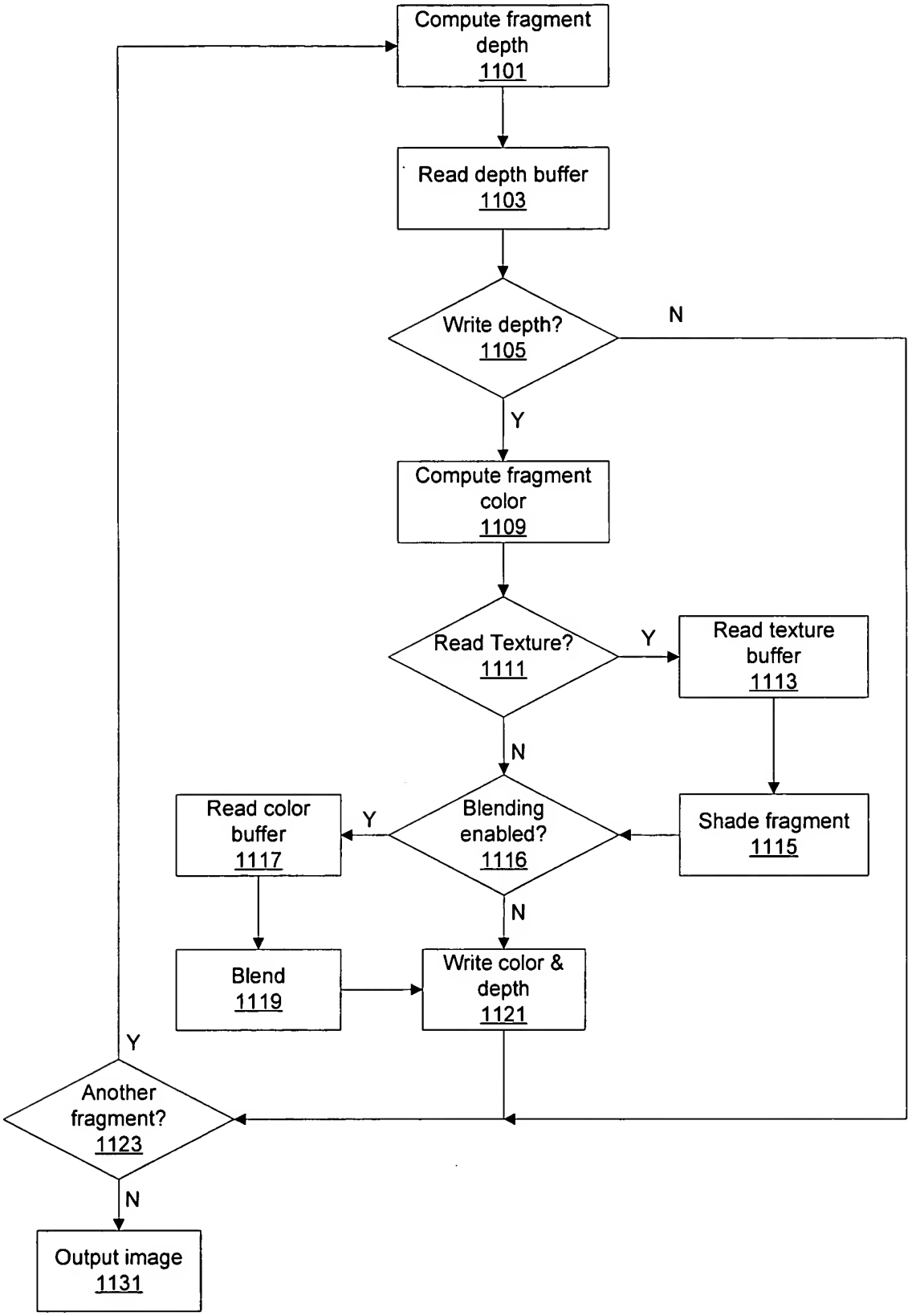
Compute fragment depth 1101
Read depth buffer 1103
Write depth? 1105
N
Y
Compute fragment color 1109
Read Texture? 1111
Y
Read texture buffer 1113
N
Read color buffer 1117
Y
Blending enabled? 1116
Shade fragment 1115
N
Blend 1119
Write color & depth 1121
Y
Another fragment? 1123
N
Output image 1131

Fig. 11A

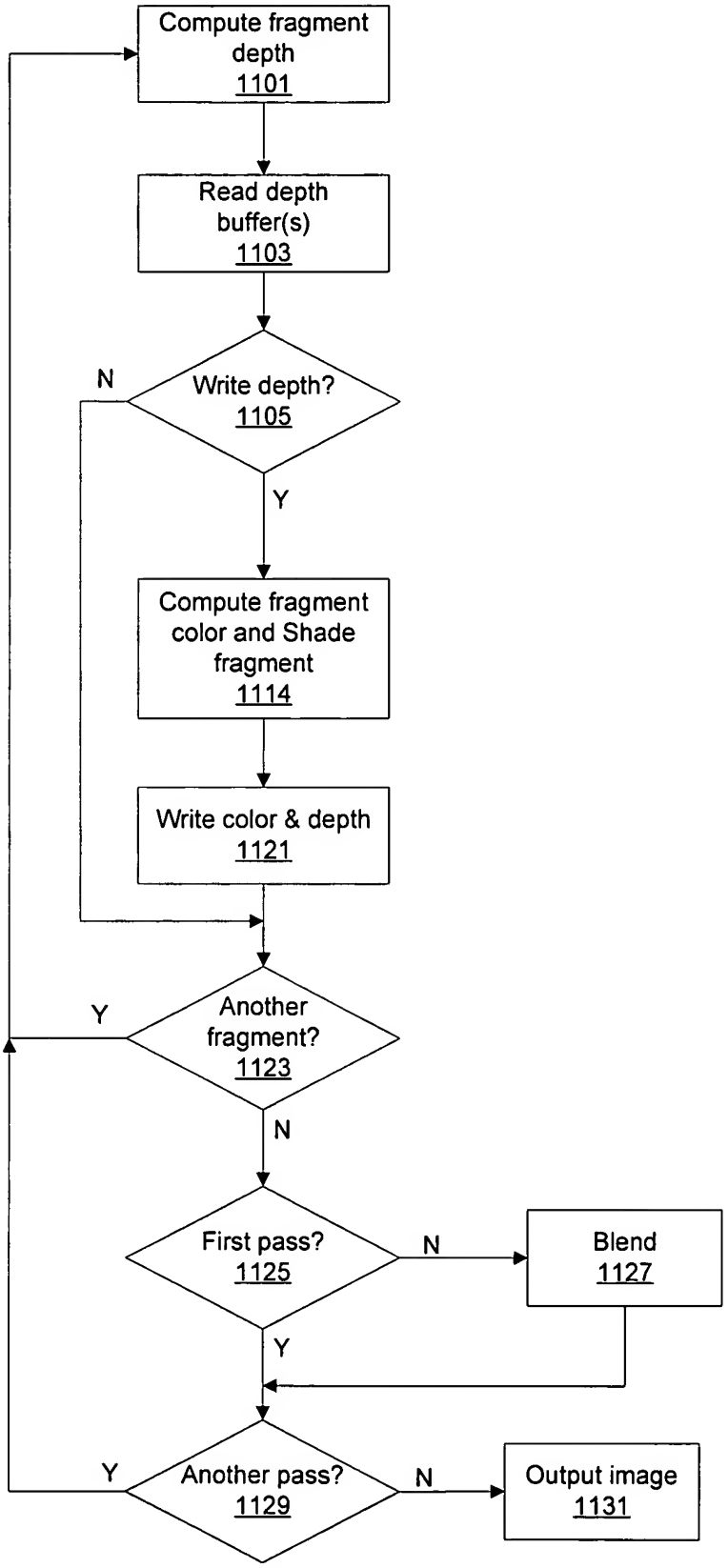
Compute fragment depth
1101
Read depth buffer(s)
1103
Write depth?
1105
N
Y
Compute fragment color and Shade fragment
1114
Write color & depth
1121
Another fragment?
1123
Y
N
First pass?
1125
N
Blend
1127
Y
Another pass?
1129
Y
N
Output image
1131

Fig. 11B

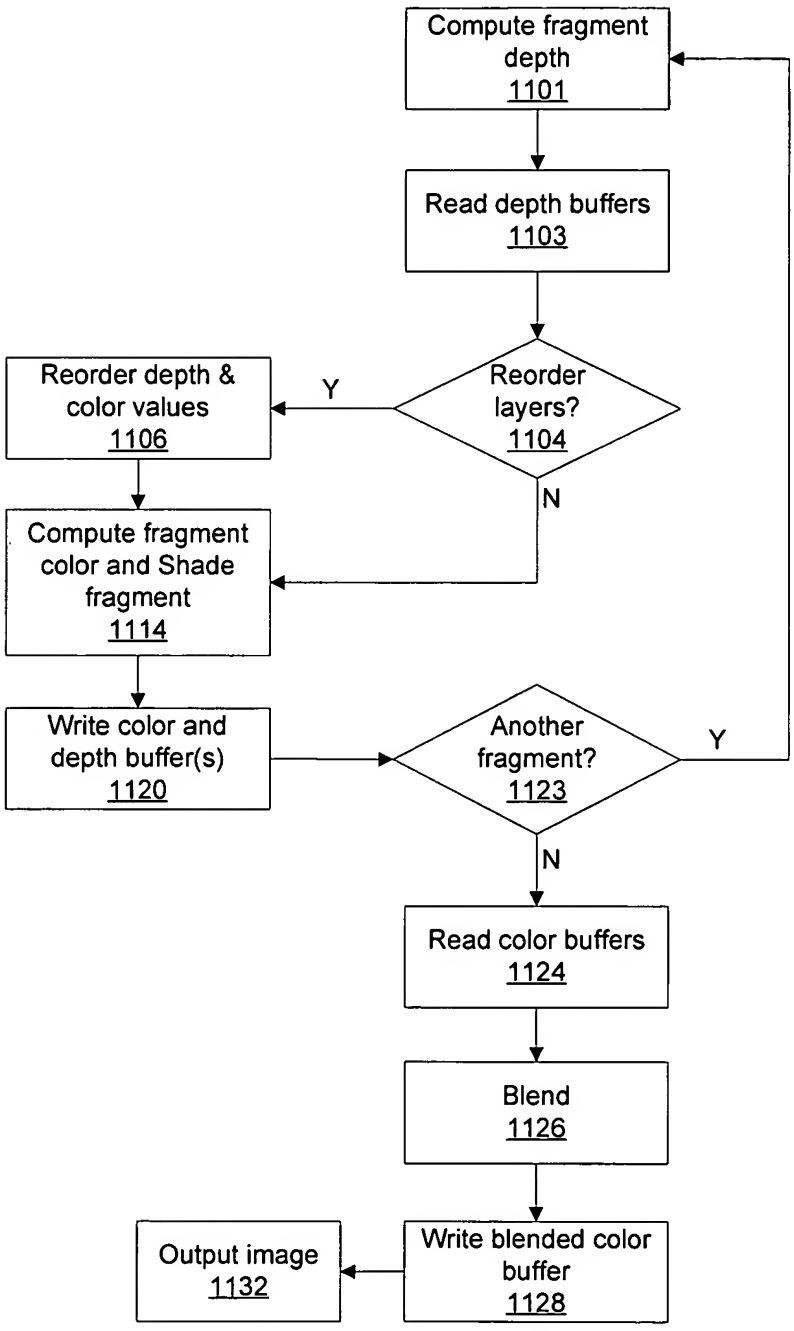
Compute fragment depth 1101
Read depth buffers 1103
Reorder layers? 1104
Y
Reorder depth & color values 1106
N
Compute fragment color and Shade fragment 1114
Write color and depth buffer(s) 1120
Another fragment? 1123
Y
N
Read color buffers 1124
Blend 1126
Write blended color buffer 1128
Output image 1132

Fig. 11C

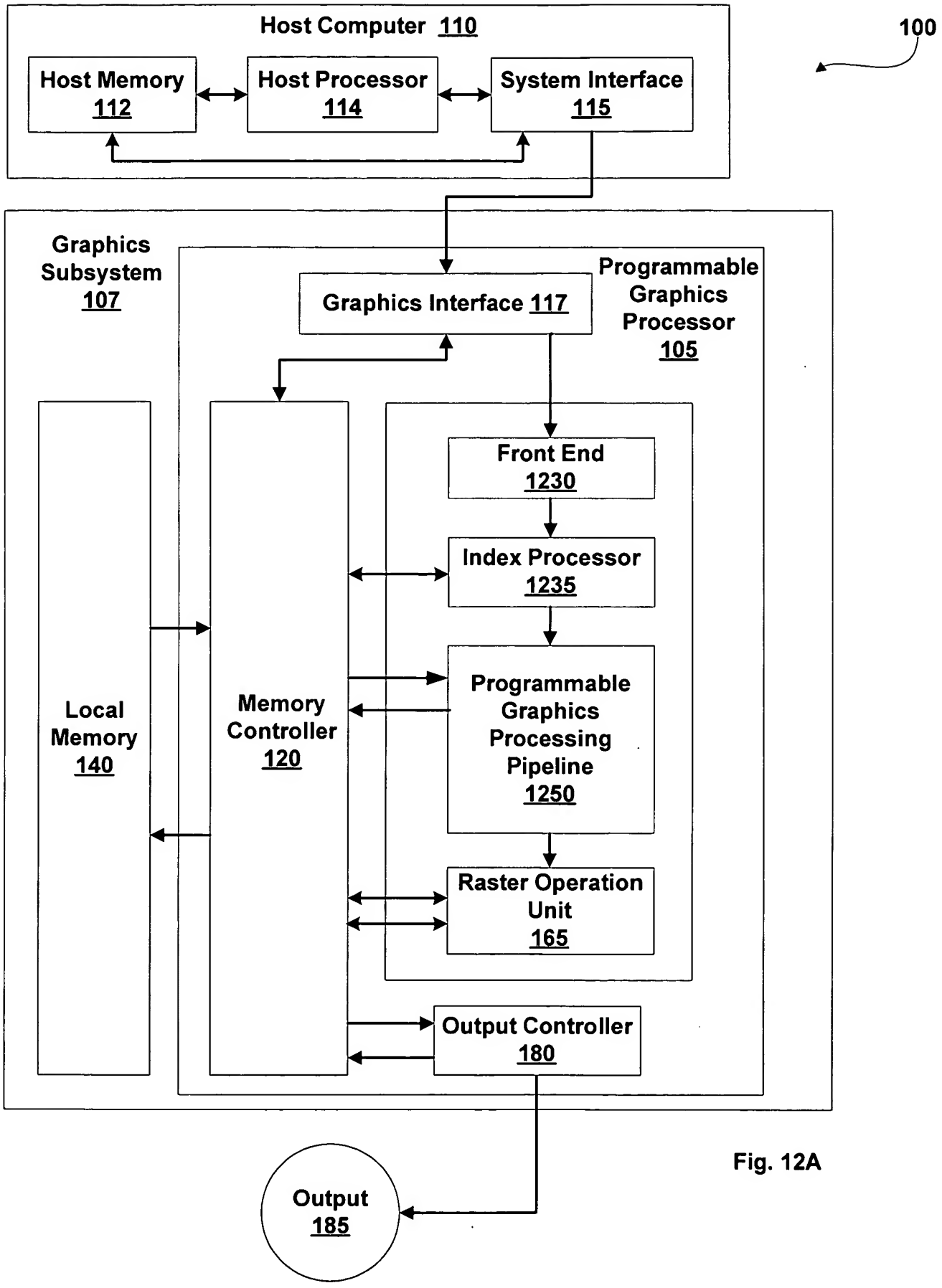
100
Host Computer 110
Host Memory 112
Host Processor 114
System Interface 115
Graphics Subsystem 107
Graphics Interface 117
Programmable Graphics Processor 105
Front End 1230
Index Processor 1235
Programmable Graphics Processing Pipeline 1250
Raster Operation Unit 165
Local Memory 140
Memory Controller 120
Output Controller 180
Output 185

Fig. 12A

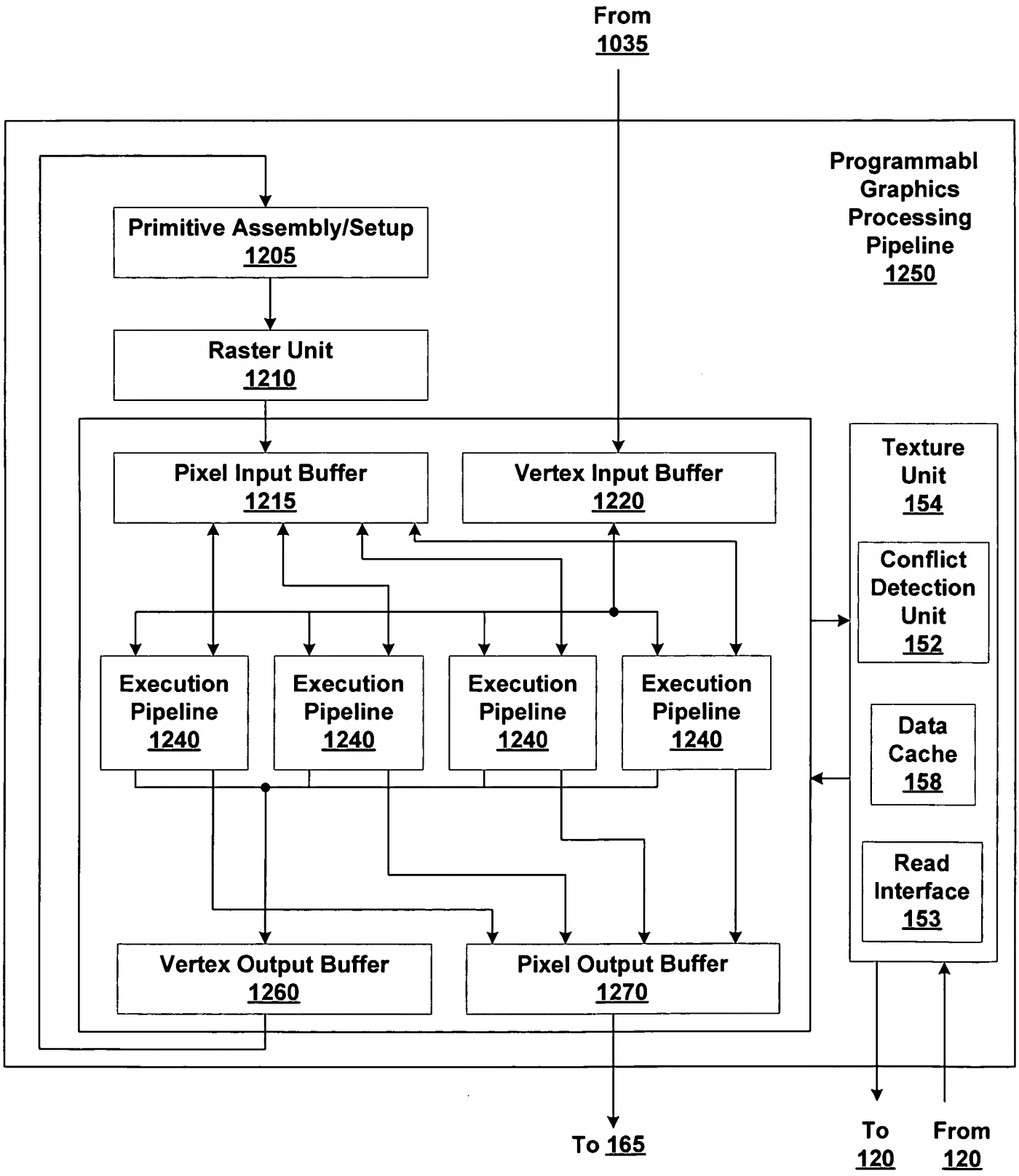
From
1035
Programmabl
Graphics
Processing
Pipeline
1250
Primitive Assembly/Setup
1205
Raster Unit
1210
Pixel Input Buffer
1215
Vertex Input Buffer
1220
Texture
Unit
154
Conflict
Detection
Unit
152
Execution
Pipeline
1240
Execution
Pipeline
1240
Execution
Pipeline
1240
Execution
Pipeline
1240
Data
Cache
158
Read
Interface
153
Vertex Output Buffer
1260
Pixel Output Buffer
1270
To 165
To
120
From
120

Fig. 12B

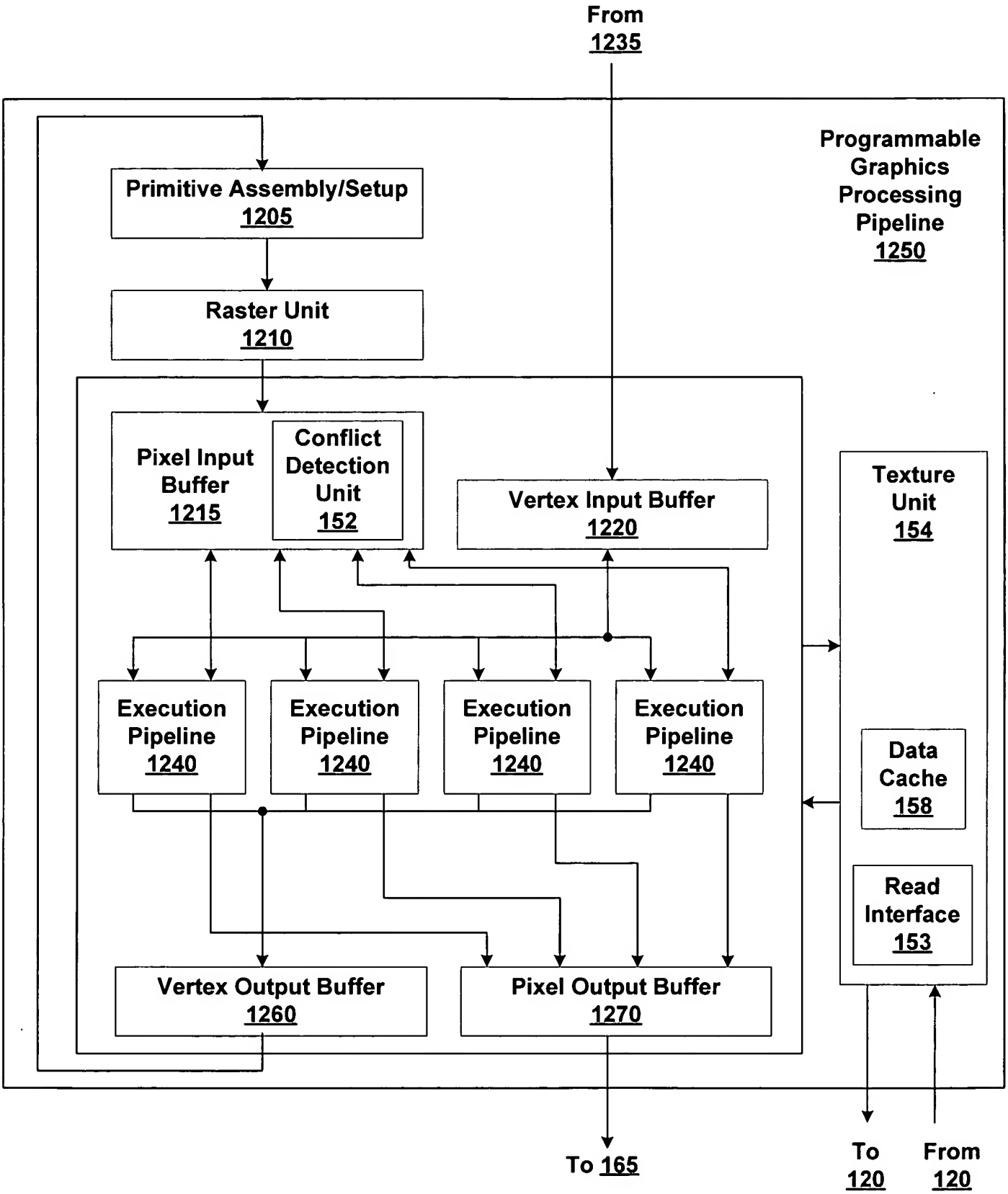
From
1235
Programmable
Graphics
Processing
Pipeline
1250
Primitive Assembly/Setup
1205
Raster Unit
1210
Pixel Input
Buffer
1215
Conflict
Detection
Unit
152
Vertex Input Buffer
1220
Texture
Unit
154
Execution
Pipeline
1240
Execution
Pipeline
1240
Execution
Pipeline
1240
Execution
Pipeline
1240
Data
Cache
158
Read
Interface
153
Vertex Output Buffer
1260
Pixel Output Buffer
1270
To 165
To
120
From
120

Fig. 12C